# Graph Representation Learning: Self-Supervision and Generative Modelling

## NORA Summer School on Geometric Deep Learning

**Raghav**endra Selvan

Assistant Professor (TT)
Dept. of Computer Science (ML Section)
University of Copenhagen
Pioneer Centre for Artificial Intelligence, Denmark

raghav@di.ku.dk
𝕏 /raghavian
`https://raghavian.github.io`

June 12, 2024

# Graph Representation Learning: Overview

1. Lecture-3
   - Graph Representation Learning
   - Self-Supervised GNNs
   - Generative Modelling for Graphs
     - Variational Autoencoders

# Representation Learning with GNNs: Overview

# Representation Learning is Compact Feature Learning

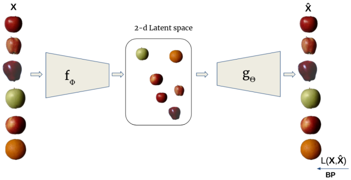
## Formalizing representation learning:

> *"Learning representations of the data that make it easier to extract useful information when building classifiers or other predictors." [1]*

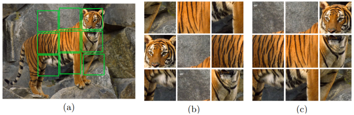
$$\mathbf{h} = g_\phi(\mathbf{z}) \quad (1)$$

$$\mathbf{z} = f_\theta(\mathbf{x}) \in \mathbb{R}^F \quad (2)$$

**h** are usually psuedo-labels, **z** can then be used in any downstream task to predict, **y**.




Undercomplete Autoencoder



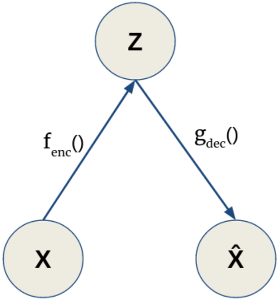

Also, interesting for graphs:

- Access meaningful vector representations of graphs
- Many applications: Graph similarity, graph matching, graph generation

[1]Representation Learning: A Review and New Perspectives. Yoshua Bengio, Aaron Courville, Pascal Vincent. 2014
Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles. Mehdi Noroozi, Paolo Favaro. 2017

# Autoencoders

- PCA is a linear dimensionality reduction method
- Autoencoders: (possibly) Non-linear PCA
- Neural Network based comprising encoder-decoder pair
- Undercomplete, Regularized, Sparse, Denoising AEs
- Compression, dimensionality reduction




Graphical model view of Autoencoders

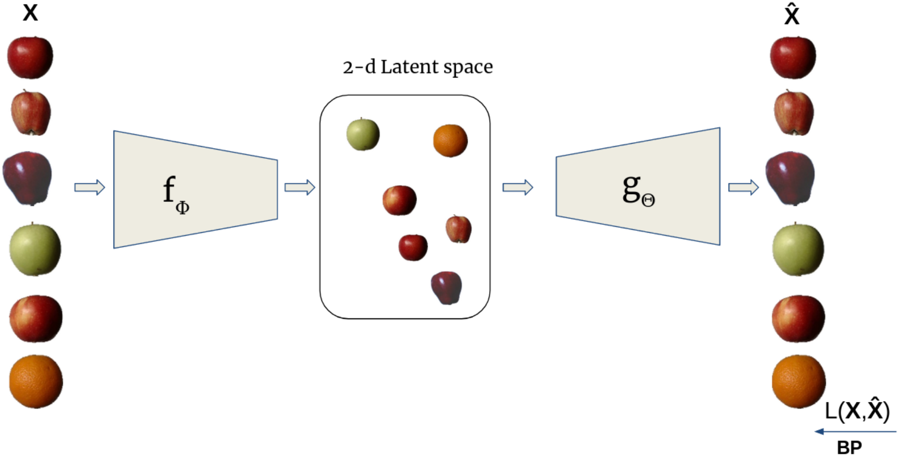
# Autoencoders

Encoder: $f_\phi(\cdot) : \mathbf{x} \in \mathbb{R}^F \rightarrow \mathbf{z} \in \mathbb{R}^D$

Decoder: $g_\theta(\cdot) : \mathbf{z} \in \mathbb{R}^D \rightarrow \hat{\mathbf{x}} \in \mathbb{R}^F$,

- Undercomplete Autoencoders: $\mathcal{L}\big(\mathbf{x}, g(f(\mathbf{x}))\big)$
  where $\mathcal{L}(\cdot)$ is a loss function penalizing $g(f(\mathbf{x})) \neq \mathbf{x}$
- Regularized Autoencoders: $\mathcal{L}\big(\mathbf{x}, g(f(\mathbf{x}))\big) + \Omega(\mathbf{z})$
  where $\Omega(\mathbf{z})$ is a regularization penalty
- Denoising Autoencoders: $\mathcal{L}\big(\mathbf{x}, g(f(\tilde{\mathbf{x}}))\big)$
  where $\tilde{\mathbf{x}}$ is $\mathbf{x}$ corrupted with some form of *stochastic* noise

# Autoencoders in practice



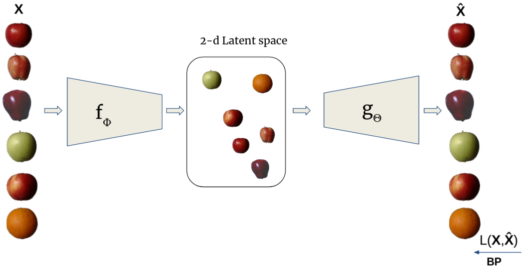

Undercomplete Autoencoder

# Optimizing the Autoencoder: An MLE perspective

- Likelihood for Autoencoders: $p(\hat{\mathbf{X}}|\mathbf{X};\theta,\phi)$
- $\theta,\phi$ are optimized to maximize the likelihood
- Maximum likelihood estimation (MLE)
- Continuous values: Mean Squared Error is commonly used
- Categorical values: Cross entropy loss

# Applying Autoencoders

- Undercomplete autoencoders $D < F$
- Non-linear low dimensional representation
- Clustering, Compression
- Feature extraction
- Similarity of data points
- Snapshot of the data *manifold*




What else can the latent space be used for?

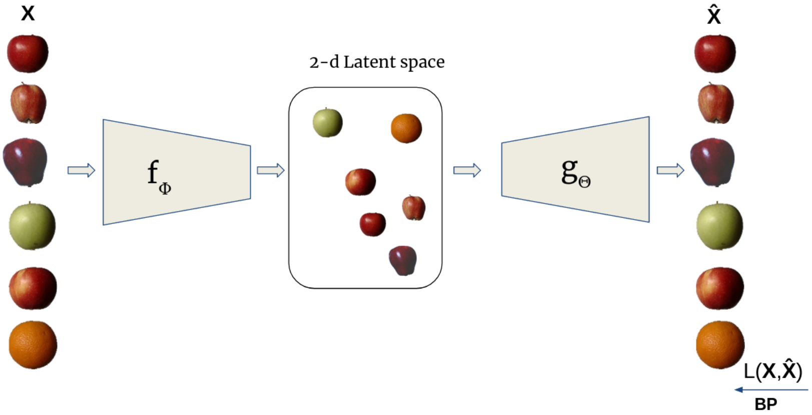
# Regularizing Autoencoders

- Latent space can be unstructured
- Discontinuities in the space
- Forcing structure with regularization
- Stochastic Autoencoders

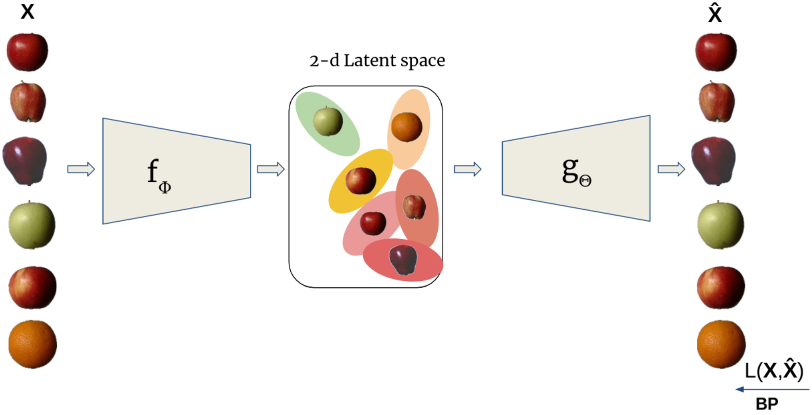
**Idea**

Instead of embedding data as points, embed them as probability densities

# Regularizing the latent space of Autoencoders



# Regularizing the latent space of Autoencoders



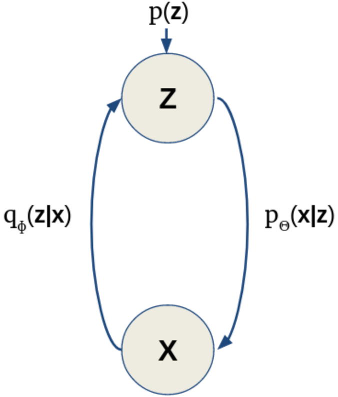
# Variational Autoencoders (VAE)

- Regularized stochastic Autoencoders
- Inspired and driven by Approximate Variational Inference
- Encoder and decoder are probabilistic
- Data is mapped to a probability density
- Deep **Generative** model

# VAE Objective

As we are in probabilistic setting:

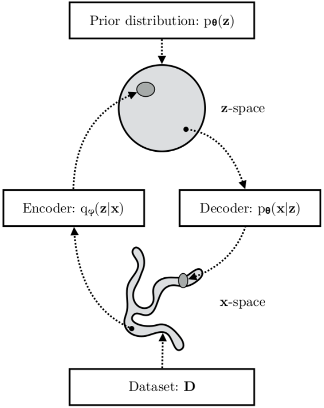
- We can estimate the data distribution $p(\mathbf{x})$
- Estimation from observed data $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$
- Probabilistic Encoder: $q_\phi(\mathbf{z}|\mathbf{x}) \approx p(\mathbf{z}|\mathbf{x})$
- Probabilistic Decoder: $p_\theta(\mathbf{x}|\mathbf{z})$
- Prior on latent variables: $p(\mathbf{z})$



# VAE objective

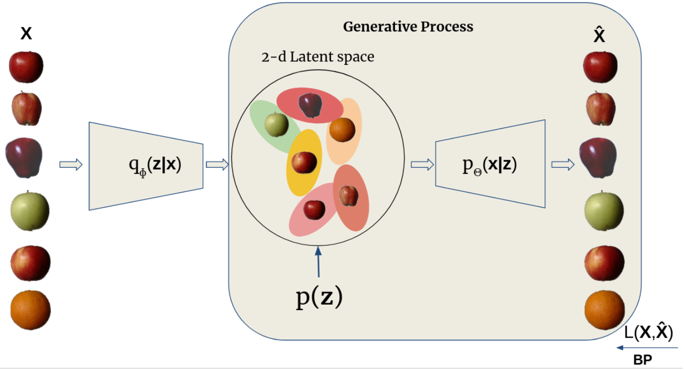
$$\mathcal{L}_{\text{VAE}} = -\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\Big[\log p_\theta(\mathbf{x}|\mathbf{z})\Big] + \text{KL}\Big[q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})\Big]$$

- Autoencoder with regularisation
- First term is reconstruction loss
- Second term is regularisation penalty
- Regularisation forces the encoder to match the prior $p(\mathbf{z})$



# Generative modelling using VAEs

- Primary difference between Autoencoders
- Exploiting structure of latent space
- Sample from prior → Decode samples
- Once trained, VAEs can be used to generate data



# Self-Supervised GNNs

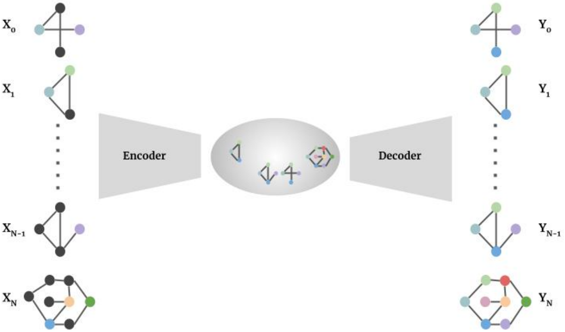
- Pseudo-labels derived graph data
- Labels can be derived from: $\mathbf{X} \in \mathbb{R}^{N \times F}, \mathbf{A} \in [0, 1]^{N \times N}, \mathbf{E} \in \mathbb{R}^{E \times N \times N}$
  - Masking node features: $\mathbf{h} = \mathbf{M} \cdot \mathbf{X}, \mathbf{M} \in \{0, 1\}^{N \times F}$
  - Noisy edges: $\mathbf{h} = 0.5 \cdot \mathbf{A} + \mathbf{N}, \mathbf{N} \in [-0.5, 0.5]^{N \times N}$
- Reconstruction with undercomplete autoencoders

# Self-Supervised GNNs

- Pseudo-labels derived graph data
- Labels can be derived from: $\mathbf{X} \in \mathbb{R}^{N \times F}, \mathbf{A} \in [0,1]^{N \times N}, \mathbf{E} \in \mathbb{R}^{E \times N \times N}$
  - Masking node features: $\mathbf{h} = \mathbf{M} \cdot \mathbf{X}, \mathbf{M} \in \{0,1\}^{N \times F}$
  - Noisy edges: $\mathbf{h} = 0.5 \cdot \mathbf{A} + \mathbf{N}, \mathbf{N} \in [-0.5, 0.5]^{N \times N}$
- Reconstruction with undercomplete autoencoders

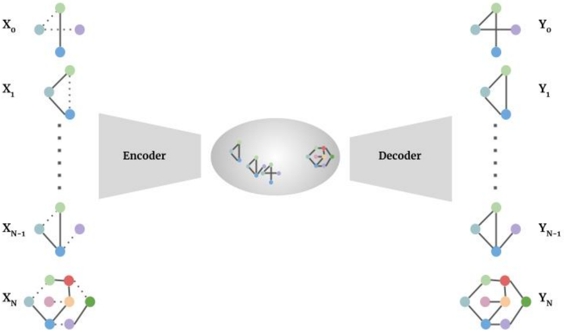
# Self-Supervised GNNs

- Pseudo-labels derived graph data
- Labels can be derived from: $\mathbf{X} \in \mathbb{R}^{N \times F}, \mathbf{A} \in [0,1]^{N \times N}, \mathbf{E} \in \mathbb{R}^{E \times N \times N}$
  - Masking node features: $\mathbf{h} = \mathbf{M} \cdot \mathbf{X}, \mathbf{M} \in \{0,1\}^{N \times F}$
  - Noisy edges: $\mathbf{h} = 0.5 \cdot \mathbf{A} + \mathbf{N}, \mathbf{N} \in [-0.5, 0.5]^{N \times N}$
- Reconstruction with undercomplete autoencoders



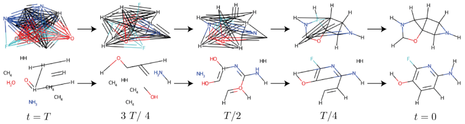
# Self-Supervised GNNs

- Pseudo-labels derived graph data
- Labels can be derived from: $\mathbf{X} \in \mathbb{R}^{N \times F}, \mathbf{A} \in [0,1]^{N \times N}, \mathbf{E} \in \mathbb{R}^{E \times N \times N}$
  - Masking node features: $\mathbf{h} = \mathbf{M} \cdot \mathbf{X}, \mathbf{M} \in \{0,1\}^{N \times F}$
  - Noisy edges: $\mathbf{h} = 0.5 \cdot \mathbf{A} + \mathbf{N}, \mathbf{N} \in [-0.5, 0.5]^{N \times N}$
- Reconstruction with undercomplete autoencoders



# Summary: Generative Modeling for Grapsh

+ Probabilistic Autoencoder
+ Latent space regularisation with prior
+ Structure of latent space exploited for generative sampling
+ Interpolation in latent space
+ Feature disentanglement
- Difficult to train for high dimensional data
- Not obvious for graph structured data
- Gaussian approximations

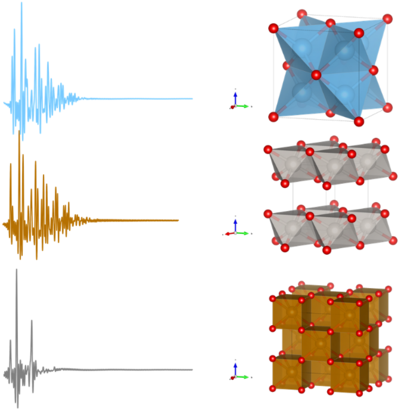



Vignac et al. 2023 DiGress: Discrete Denoising diffusion for graph generation

# Representation Learning with GNNs: Overview

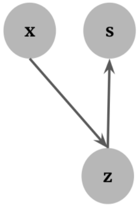
1. Lecture-3
   - Graph Representation Learning
   - Self-Supervised GNNs
   - Generative Modelling for Graphs
     - Variational Autoencoders

# GNNs for Characterising Atomic Structure of Mono-Metallic Nanoparticles



- Solve structures starting from X-ray scattering measurements
- Reconstruct structures of nanoparticles, $\mathbf{s} \in \mathcal{S}$, from their corresponding property (PDFs), $\mathbf{x} \in \mathcal{X}$.
- Learning task: $f(\cdot) : \mathcal{X} \rightarrow \mathcal{S}$.
- From a density esimtation point of view: $p(\mathbf{s}|\mathbf{x})$.
- Many-to-one mapping

# Latent Generative Model

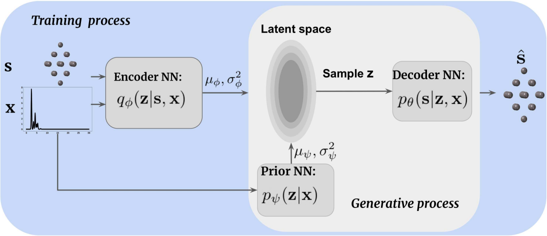



conditional-VAE

- Treat structures as a graphs with atoms as nodes, distances as edge attributes
- Using property-structure pairs formulate a conditional generative model
- Conditional Variational Autoencoder (CVAE)
- CVAE extends VAE framework to include conditioning input
- CVAE objective minimizes KLD between $p(\mathbf{z}|\mathbf{s}, \mathbf{x})$ and its variational approximation $q_\phi(\mathbf{z}|\mathbf{s}, \mathbf{x})$ resulting in an objective of the form:

$$\begin{aligned} \mathcal{L}_{\text{CVAE}} &= \mathcal{L}_{sup} + \mathcal{L}_{reg} && (3) \\ &= -\mathbb{E}_{q_\phi}\big[\log p_\theta(\mathbf{s}|\mathbf{z}, \mathbf{x})\big] + \text{KL}\big[q_\phi(\mathbf{z}|\mathbf{s}, \mathbf{x})||p_\psi(\mathbf{z}|\mathbf{x})\big] && (4) \end{aligned}$$

# High level overview of the conditional generative model




- Conditioning input at Encoder/Prior networks using MLPs
- Encoder only during training
- Inference using Prior network alone
- Use GNNs in the encoder $q_\phi(\cdot)$:

$$\mathbf{H}^{(m)} = \sigma(\mathbf{H}^{(m-1)}, \mathbf{A}; \Theta_{m-1}) \quad (5)$$

with $\mathbf{H}^{(0)} = \mathbf{X} \in \mathbb{R}^{N\times F}$ and $\mathbf{H}^{(M)} = \mathbf{Z} \in \mathbb{R}^{N\times L}$

- Trained entirely with simulated data

Anker.. Selvan et al. 2021

# Joint Representation Space of property + structures in 2D latent space

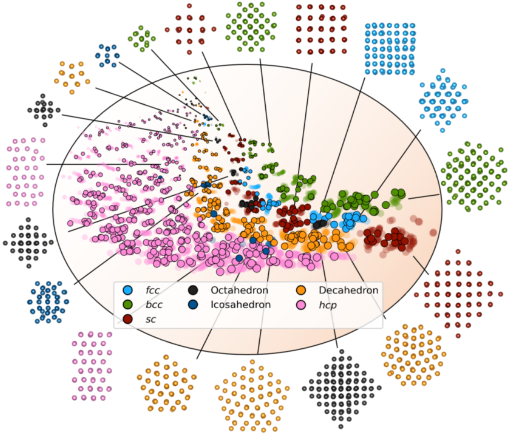



**Fig. 2 | The two-dimensional latent space with structure reconstructions.** The points in the latent space

Anker .. Selvan et al. 2023, DeepStruc

# Meaningful interpolation in latent space

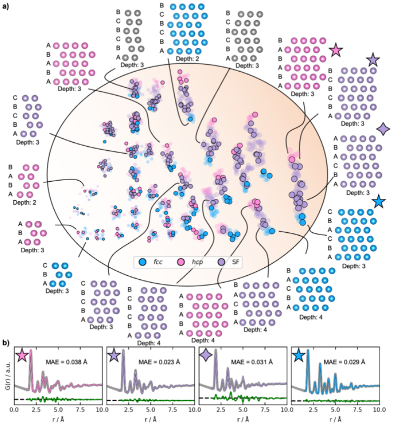



**Fig. 6 | Latent space and reconstructions of stacking faulted nanoparticles.** a) The latent space and reconstructed structures shown with their stacking sequence. The structures are shown in two dimensions, and

---

Anker .. Selvan et al. 2023, DeepStruc

# Results on structure prediction of nanoparticles based on properties

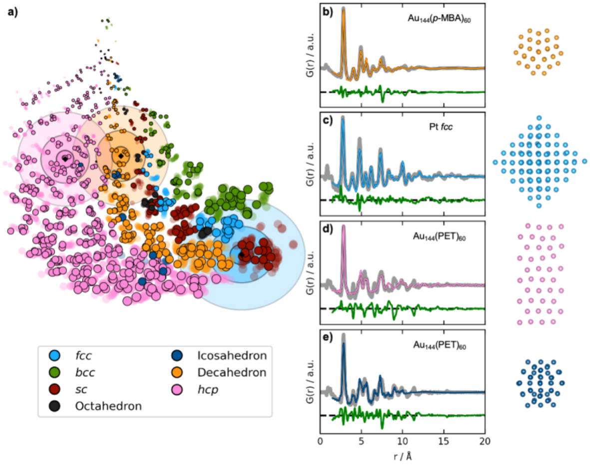



**Fig. 5 | Fitting experimental PDFs with structures obtained by DeepStruc.** a) The DeepStruc latent space